

Práctica de la  
Asignatura de **Programación III**  
Versión 1.0

Escuela Técnica Superior de  
Ingeniería Informática

**UNED**

**Curso 2009-2010**

## 1.- ENUNCIADO DE LA PRÁCTICA

*Hashiwokakero* es un puzzle lógico original de Nikoli. En su web se pueden encontrar las instrucciones originales (<http://www.nikoli.co.jp/en/puzzles/hashiwokakero/>).

El pasatiempo consiste en un tablero rectangular, aunque el tablero en sí no se suele dibujar, en el que algunas celdas contendrán números con valores del 1 al 8, que aparecen dentro de un círculo y que representan “islas”. El resto de las celdas estarán vacías. El objetivo es conectar todas las islas dibujando “puentes” (aristas) entre las islas. Los puentes deben cumplir las siguientes condiciones:

- Deben comenzar y acabar en distintas islas mediante una línea recta.
- No pueden cruzar otros puentes u otras islas.
- Las líneas rectas deben ser horizontales o verticales.
- Como mucho puede haber dos puentes paralelos que conecten un par de islas.
- Al final, el número de puentes conectados a cada isla debe coincidir con el número de la isla.

La siguiente figura muestra un ejemplo del juego:

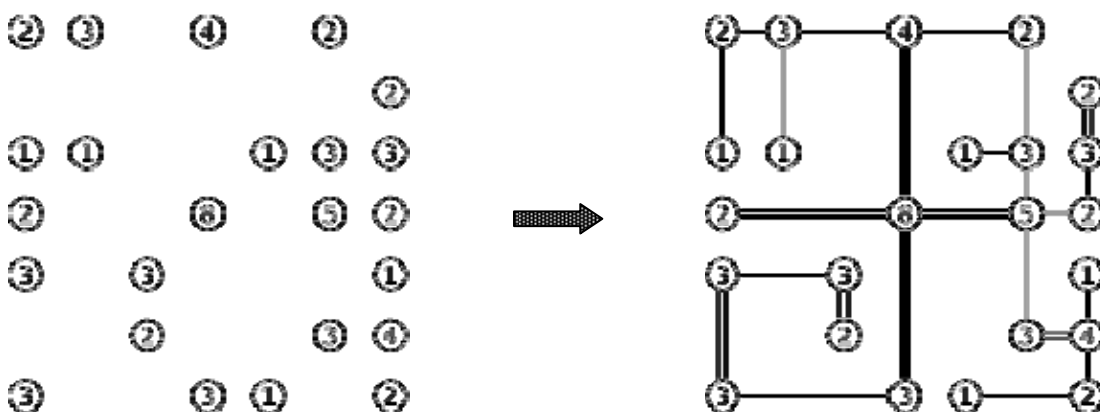


Figura 1: Un ejemplo de *Hashiwokakero* (<http://en.wikipedia.org/wiki/Hashiwokakero>)

Dado un tablero de *Hashiwokakero* con las islas colocadas, la práctica consiste en hallar una solución al puzzle, si la hay, de manera que todas las islas estén interconectadas por puentes, es decir, que el tablero (grafo) resultante sea conexo.

## 2.- REALIZACIÓN DE LA PRÁCTICA

La práctica constará de una memoria y de un programa en Java que resuelva el problema planteado mediante el esquema más apropiado de entre los expuestos en la asignatura.

En la memoria se detallarán las respuestas a las preguntas planteadas y un listado

comentado del código fuente. Se valorará que el código esté comentado de forma que se facilite su lectura. La práctica se realizará en lenguaje **Java**.

## 2.1.- Diseño del Algoritmo

Contesta y desarrolla las siguientes preguntas:

1. Describe el tipo de problema y justifica el esquema algorítmico utilizado.
2. Expón las estrategias locales consideradas en la asignación de valores. Concrétalas en condiciones de poda.
3. Analiza el coste en términos de tratabilidad computacional.

Orientaciones:

- Sólo hay 4 direcciones posibles para los puentes (arriba, abajo, derecha e izquierda). Si una isla tiene valor '8', deberá tener los 2x4 posibles puentes y es un buen punto de comienzo. De la misma manera, si una isla está en una esquina y tiene un '4', o está en a lo largo del borde y tiene un '6' también debe tener 2 puentes en cada dirección. Cualquiera de esas islas nos permiten tratar puentes que deberían estar en la solución. A partir de ellos se puede ir construyendo la solución.
- Se pueden ir "tachando" las islas que ya contienen todos sus puentes.
- Se recomienda encontrar una solución que satisfaga la condición del número de puentes de las islas, y posteriormente comprobar si dicha solución permite que todas las islas estén conectadas entre sí.
- Para comprobar que todas las islas están conectadas entre sí basta hacer un recorrido en anchura o profundidad y comprobar si se han "visitado" todos los nodos. Para ello habrá que transformar el tablero en un grafo en el que en el que los nodos serían las islas y los puentes las aristas.

## 2.2.- Funcionalidad de la práctica

### 2.2.1.- Argumentos y parámetros

El programa se deberá poder invocar **desde la línea de comandos** de Linux (ó Windows) con la siguiente sintaxis:

```
$> java hashiwokakero [-h] [fichero]
```

Los argumentos son:

fichero: el nombre del archivo que contiene los datos de entrada.  
-h: modo ayuda. Muestra la sintaxis y los créditos.

Si no tiene argumentos, el programa muestra la sintaxis.

El programa leerá los datos del fichero que se le pase como argumento. Por ejemplo:

```
C:\> java hashiwokakero tablero.txt <ENTER>
```

o también desde la entrada estándar (stdin):

```
C:\> java hashiwokakero < tablero.txt <ENTER>
```

o también usando *pipe* (*tubería*)

```
C:\> type tablero.txt | java hashiwokakero <ENTER>
```

y saca por la salida estándar el tablero solución o una indicación de que no hay solución si ese es el caso.

Los programas que no permitan esta funcionalidad estarán suspensos.

### 2.2.2.- Formato de la entrada y salida de datos

El fichero de entrada representará el tablero inicial del juego. El tablero se representa por filas de la siguiente forma:

- Cada línea del fichero es una fila del tablero excepto la primera y la segunda. La primera línea define el número de filas del tablero y la segunda línea el número de columnas.
- Si la celda contiene un número el fichero contendrá dicho número.
- Los huecos se representan mediante el carácter '\*' (asterisco).
- El carácter separador de celdas es un espacio (blanco).
- Un salto de línea representa el fin de una fila.
- Se ignorará todas las filas restantes a partir de que se encuentre el carácter '+', que indicará el fin de la descripción del tablero.

Por ejemplo (Linux) se muestra un fichero correspondiente al *hashiwokakero* de la Figura 1:

```
$> cat tablero.txt <ENTER>

#-----
# Ejemplo hashiwokakero
# Fuente: Nikoli
# Valores de la Figura 1
```

```
# -----
##
13
13
2 * 3 * * * 4 * * * 2 * *
* * * * * * * * * * * *
* * * * * * * * * * * * 2
* * * * * * * * * * * *
1 * 1 * * * * * 1 * 3 * 3
* * * * * * * * * * * *
2 * * * * * 8 * * * 5 * 2
* * * * * * * * * * * *
3 * * * 3 * * * * * * 1
* * * * * * * * * * * *
* * * * 2 * * * * * 3 * 4
* * * * * * * * * * * *
3 * * * * * 3 * 1 * * * 2
+
$>
```

El tablero de salida se puede mostrar de la siguiente manera:

- Una isla en una celda se representa por su número.
- Un puente en una celda se representa por los caracteres “-” o “|” dependiendo de si es un puente horizontal o vertical. En caso de doble puente se representará por el carácter “=” en horizontal y el carácter “H” en vertical.
- La no existencia ni de puente ni de celda se representa con un espacio en blanco.

Un ejemplo de ejecución (Windows) a partir del fichero anterior podría ser:

```
C:\> type tablero.txt | java hashiwokakero <ENTER>
```

Existe solución que conecta todas las islas:

```
2 - 3 - - - 4 - - - 2
|         |         H         |
|         |         H         | 2
|         |         H         | H
1   1         H   1 - 3   3
          H         |   |
2 = = = = = 8 = = = 5 - 2
          H         |
3 - - - 3   H         |   1
H         H   H         |   |
H         2   H         3 = 4
H         H         |
3 - - - - - 3   1 - - - 2

C:\>
```

### 3.- IMPLEMENTACIÓN

#### 3.1.- Compiladores

##### Compilador de Java

Se usará Java 1.4 o superior. Todo lo necesario está disponible en <http://java.sun.com>. Cualquier entorno de desarrollo es útil para realizar la práctica. Basta en cualquier caso un editor de línea aunque se recomienda BlueJ (<http://www.bluej.org>)

#### 3.2.- Estudio del Coste

Estudiar y calcular si es posible la complejidad del algoritmo. Realizar un estudio teórico del coste del algoritmo implementado y describir las condiciones de poda realizadas.

#### 3.3.- Casos de Prueba

Usar como caso de prueba el ejemplo inicial de la Figura 1.

#### 3.4.-Diseño del programa y estructuras de datos

Se recomienda que el diseño del programa siga un enfoque orientado a objetos pero no será necesario usar el concepto de herencia. Se entiende que lo aprendido en Programación I y II sobre diseño modular se debe aplicar a la práctica.

Para las estructuras de datos se usarán las proporcionadas por Java.

### 4.- ENTREGA DE LA PRÁCTICA

1. La asistencia a las sesiones de prácticas es obligatoria **SIN** excepciones.
2. El calendario y procedimiento para asistir a las sesiones de prácticas **está publicado en su Centro Asociado.**
3. El plazo de entrega de la documentación y de la práctica **lo establece el Tutor de prácticas de cada Centro Asociado.**
4. El Tutor califica la práctica, informa al alumno y en su caso la revisa de acuerdo con los horarios y procedimiento que establezca el Centro Asociado.
5. La Sede Central **NO** informa de las calificaciones de las prácticas, lo hace el Tutor o el Centro Asociado.

6. TODOS los alumnos deberán registrarse a través del Curso Virtual (acceso desde CiberUNED) con el Tutor/a con el que hayan asistido a las sesiones presenciales obligatorias a fin de que su práctica pueda ser calificada.
7. No se recogerá documentación alguna de alumnos que no hayan asistido a las sesiones obligatorias.
8. La práctica se debe aprobar en la misma o anterior convocatoria para que se pueda calificar la asignatura. En caso contrario la calificación será de suspenso.
9. La práctica se entregará empaquetada en un \*.jar tanto en el entorno WebCT como al Tutor. La falta de cualquiera de ellas será motivo suficiente para quedar excluida de la convocatoria.

## TUTORES

1. La práctica se califica mediante un entorno web accesible desde los Cursos Virtuales.
2. Los tutores se podrán de alta en la aplicación de entrega de notas de la práctica, sin necesidad de ponerse en contacto con el equipo docente. En el Foro de tutores recibirán las instrucciones pertinentes.
3. Los alumnos se dan de alta en la aplicación de corrección y calificación de prácticas accesible desde WebCT de acuerdo con las indicaciones del Tutor. Aquel alumno que no se dé de alta no aparecerá en la lista de presentados a la misma y no constará su nota en el entorno de calificación y revisiones.

## Material que hay que entregar al Tutor

Se confeccionará una memoria escrita con índice:

1. Respuesta a las cuestiones planteadas en este enunciado.
2. Un ejemplo de ejecución para el caso de prueba.
3. Estudio del coste del algoritmo.
4. Un listado del código fuente COMPLETO.

Se adjuntará un CD con el nombre y apellidos del alumno escrito en el mismo y conteniendo:

1. Fuentes de cada uno de los módulos.
2. Ejecutable \*.jar del programa principal.
3. Ficheros de los casos de prueba.
4. Memoria de la práctica en Word, LaTeX o OpenOffice.

El disco y toda la documentación que se entregue deberá tener bien legible: **Nombre y apellidos, C. Asociado, teléfono y e-mail.**

El alumno debe asegurarse de que no se da ninguna de las siguientes circunstancias, ya que implican automáticamente una calificación de SUSPENSO:

- **Código:** El código no compila, no está desarrollado en Java, no se corresponde con el pseudocódigo recogido en la documentación, no es original, está copiado de la red, academia, compañero, etc., o no sigue un diseño OO encapsulado o modular.
- **Ejecutable:** El ejecutable no termina, se queda sin memoria con ejemplares pequeños o aborta sin justificación. El ejecutable no lee los ficheros previstos en el formato adecuado. No trata los argumentos o no se ajusta a las especificaciones.
- **Documentación:** No se presenta en papel o está incompleta.
- **Soporte:** No se puede leer, o contiene un virus de cualquier tipo. A este respecto, las prácticas en las que se detecte cualquier tipo de virus estarán suspensas.

### Entrega en WebCT

Se entregará en el entorno WebCT como trabajo la misma documentación entregada al tutor, de acuerdo con las instrucciones que se pondrán en el entorno. Esta entrega es **obligatoria**. Las prácticas se procesarán automáticamente mediante un software de tratamiento masivo para detectar copias. Los alumnos con prácticas copiadas estarán suspensos y serán objeto de apertura de expediente disciplinario.

\*\*\*\*

**La práctica debe realizarse de forma individual y se entregará al TUTOR en las fechas que fije cada Centro Asociado.**

\*\*\*\*

El Centro Asociado publicará las notas de las prácticas. El equipo docente **NO INFORMARÁ DE LAS NOTAS DE PRÁCTICAS**, debiendo el Alumno dirigirse al Centro Asociado para conocerlas.

## **ANEXO I**

### **Bibliografía y tutoriales de Java**

La referencia más útil es el libro “*Piensa en Java*” (“*Think in Java*”) de Bruce Eckel, disponible en varias editoriales. Hay una versión completa y gratuita en la red en <http://www.mindview.net/Books/TIJ>.

En <http://www.javahispano.org/> hay multitud de tutoriales en español.

En la web de BlueJ también hay mucha información interesante.

### **Alumnos sin TUTOR**

No hay sesiones de prácticas para alumnos sin Tutor, ya que no es posible según la normativa de la UNED matricularse en aquellos Centros que no disponen de Tutor de esta asignatura.

Los alumnos estudiando en el EXTRANJERO pónganse en contacto con Antonio Sarasa en el FORO de “alumnos fuera de España” de WebCT.